

日本国特許庁
JAPAN PATENT OFFICE

CFA00040
US

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年12月16日

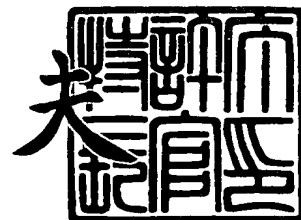
出願番号
Application Number: 特願2002-363917
[ST. 10/C]: [JP 2002-363917]

出願人
Applicant(s): キヤノン株式会社

2004年 1月14日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2003-3110927

【書類名】 特許願

【整理番号】 251105

【提出日】 平成14年12月16日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06F 17/50

【発明の名称】 最適設計方法

【請求項の数】 1

【発明者】

 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社
社内

 【氏名】 鷺澤 輝芳

【発明者】

 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社
社内

 【氏名】 浅井 朗

【発明者】

 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社
社内

 【氏名】 新庄 克彦

【発明者】

 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社
社内

 【氏名】 立原 昌義

【発明者】

 【住所又は居所】 埼玉県朝霞市幸町 2 丁目 1 1 番 3 号

 【氏名】 吉川 暢宏

【特許出願人】

【識別番号】 000001007
【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号
【氏名又は名称】 キヤノン株式会社
【代表者】 御手洗 富士夫

【代理人】

【識別番号】 100090538
【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社
内

【弁理士】

【氏名又は名称】 西山 恵三
【電話番号】 03-3758-2111

【選任した代理人】

【識別番号】 100096965
【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会
社内

【弁理士】

【氏名又は名称】 内尾 裕一
【電話番号】 03-3758-2111

【手数料の表示】

【予納台帳番号】 011224
【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9908388

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 最適設計方法

【特許請求の範囲】

【請求項 1】 設計変数ベクトルをパラメタとして状態変数ベクトルに対する第 1 の評価関数の最適化問題を解く第 1 の求解工程と、

前記第 1 の求解工程で求められた状態変数ベクトルと、前記設計変数ベクトルとに対する第 2 の評価関数の最適化問題を解く第 2 の求解工程とを備える最適設計方法において、

前記第 2 の求解工程は、

前記設計ベクトルに関する前記第 2 の評価関数の勾配ベクトルを計算する勾配ベクトル計算工程と、

前記勾配ベクトルのノルムの値に基づいて第 1 の係数を計算する第 1 の係数計算工程と、

該第 1 の係数に基づいて探索ベクトルを計算する探索ベクトル計算工程と、

第 2 の係数を計算する第 2 の係数計算工程と、

前記第 2 の係数に基づいて設計変数ベクトルを更新する設計変数ベクトル更新工程とを備え、

前記第 2 の係数計算工程は前記第 1 の求解工程を含み、該第 1 の求解工程は勾配ベクトルに基づく反復解法であり、当該反復中は前記状態変数ベクトルを初期化しないことを特徴とする最適設計方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、設計パラメタの最適化問題の解法に関し、特に構造部材のトポロジーを含む形状最適化のための自動設計技術に関するものである。

【0002】

【従来の技術】

構造トポロジー最適設計とは、与えられた条件の下で最適な、構造部材のトポロジーおよび形状寸法を決定する問題である。以下、構造部材のトポロジーおよ

び形状寸法を設計変関数といい、上記決定問題を設計変関数最適化問題という。変関数という理由は、トポロジーおよび形状寸法が3次元空間の関数になっているからである。

【0003】

設計変関数最適化問題では、各設計変関数の値に対して、状態変関数の最適化問題を解かなければならない。この意味から、構造トポロジー最適設計は、内側に状態変関数の最適化問題、外側に設計変関数の最適化問題を持つ、2重構造最適化問題と捉える事が出来る。内側の状態変関数最適化問題では、従来技術の蓄積により、空間を有限個の要素に分割するという考え方が採用される。

【0004】

特に構造部材の歪エネルギーを評価汎関数としている問題では、その解析手法として有限要素法が一般的である。有限要素法の解法としては一次方程式に対する直接法が採用されている。一方、設計変関数の最適化問題に対しては、大別すると、以下に示す3種類の方法が提供されている（〔文献1〕或いは〔文献2〕）：

1. E v o l u t i o n a r y 法（以下、E法）
2. H o m o g e n i z a t i o n 法（以下、H法）
3. M a t e r i a l d i s t r i b u t i o n 法（以下、MD法）

E法では、空間を分割することによって得られる部分空間のそれぞれをセルと称し、その生成と消滅を適当な規則によって繰り返す。構造部材は、最終的に存在しているセルの集合として与えられる。セルが存在するか否かという2つの状態のみを許すことにより、明確な構造部材が得られる。また、評価汎関数の微分情報を用いないので、局所最適解にトラップされないことから、評価汎関数が多峰性の場合に有効である。

【0005】

〔文献3〕では、E法的一种である遺伝的探索法を用いた骨組構造部材の最適化設計装置が提供されている。この最適化設計装置では、骨組部材断面寸法などの離散設計変数データの近似式を使用する近似最適化計算装置と、該設計変数データを使用する詳細最適化計算装置を設け、これら2つの計算装置を結合して、

膨大な設計変数が存在する実設計問題に対応することが出来るようにしている。

【0006】

H法は、分割された各部分空間に位置する構造要素に、更に微細な構造を仮定し、連続値を取る設計変関数を新たに導入することによって、感度解析の採用を可能にしている。ここで感度解析とは、設計変関数に関する評価汎関数の微分情報を利用した解析手法のことであり、感度解析が可能になれば、勾配法のような反復解法を用いることが出来、E法のような総当りの手法に比べて、少なくとも局所最適解の探索に掛かる計算時間は大幅に短縮される。

【0007】

MD法は、構造部材の存在確率を示す0から1の範囲の実数を各要素に割当てることによって、構造部材のトポロジーや形状寸法変化を表現する方法である。構造部材が存在するか否かという離散的な情報を存在確率という連続値に置き換えることによって感度解析を可能にしたという意味でH法と同様のものであるが、H法よりパラメタ数が少ない分、モデル化が容易であり適用範囲も広い。

【0008】

〔文献5〕にはMD法による構造物の位相最適化手法が開示されている。本手法の特徴は以下のとおりである：

(1) ボクセル有限要素法（空間を等間隔に分割）を用いているため、あらゆる要素に対する要素剛性マトリクスが同一である。従って、要素剛性マトリクスを予め1度だけ計算しておけば、以後の計算に利用できる。更に、要素が規則正しく配置されているため、各要素の節点番号情報を記憶する必要がない。

【0009】

(2) 大規模連立一次方程式を解くために、前処理付共役勾配法とElement-by-Element法を組み合わせることで、全体剛性マトリクスを組み立てることなく解が求められるので、処理に必要なメモリ容量が少なくて済む。

【0010】

(3) 均質化法では、1要素に対して6個の設計変数（3次元の場合）が必要になる。更に設計変数が変化するたびに要素剛性マトリクスを再計算しなければ

ならない。一方、構造部材の存在率を密度比で表現する密度法を採用することによって、1要素に対して1つの設計変数でよい。また設計変数の変化が要素剛性マトリクスに影響を与えない。

〔文献1〕S. Bulman, J. Sienz, E. Hinton: "Comparisons between algorithms for structural topology optimization using a series of benchmark studies", Computers and Structures, 79, pp. 1203-1218 (2001) .

〔文献2〕Y-L. Hsu, M-S. Hsu, C-T. Chen: "Interpreting results from topology optimization using density contours", Computers and Structures, 79, pp. 1049-1058 (2001) .

〔文献3〕特開2001-134628号公報

〔文献4〕山川宏: "最適化デザイン", 計算力学とCAEシリーズ9, 培風館 (1996)

〔文献5〕藤井, 鈴木, 大坪: "ボクセル有限要素法を用いた構造物の位相最適化", Transactions of JSCES, Paper No. 20000010 (2000) .

【0011】

【発明が解決しようとする課題】

しかしながら、従来技術には以下に述べるような課題があった。

【0012】

一般に構造最適化問題は、設計変数ベクトルの最適化問題の各反復ステップ毎に、状態変数ベクトルの最適化問題を含む、2重の最適化問題として定式化される。設計変数ベクトルの最適化問題を外側の最適化、状態変数ベクトルの最適化問題を内側の最適化と呼ぶと、内側の最適化は、設計変数ベクトルをパラメタとし、即ち設計変数ベクトルを固定し、状態変数ベクトルを求める問題である。こ

れは通常、構造解析と呼ばれ、有限要素法によって線形方程式の解法を用いて解く事が出来る。

【0 0 1 3】

ところが、構造が変化し、ある領域の構造部材が存在なくなると、例えば部材に穴が空いてしまうと、対応する要素の設計変数が0となり、結果的に要素のヤング率が0となる。すると、上記線形方程式の係数行列がフルランクでなくなり、逆行列が存在しないという理由で、直接法では解けなくなる。

【0 0 1 4】

そのため、殆どの従来技術では、設計変数ベクトルが0となる場合には、正確に0とはせずに0に近い微小な値で置き換えるという方法を取ってきた。〔文献5〕で採用されているボクセル有限要素法でも同様である。

【0 0 1 5】

しかるに、要素の設計変数を微小な値に設定するということは、物理的には薄い膜、或いは弱い部材が存在することに相当する。つまり従来法では、物質がない部分を正確に表現していないことになる。

【0 0 1 6】

【課題を解決するための手段】

上記課題を解決するために、本発明によれば、設計変数ベクトルをパラメタとして状態変数ベクトルに対する第1の評価関数の最適化問題を解く第1の求解工程と、前記第1の求解工程で求められた状態変数ベクトルと、前記設計変数ベクトルとに対する第2の評価関数の最適化問題を解く第2の求解工程とを備える最適設計方法において、前記第2の求解工程は、前記設計ベクトルに関する前記第2の評価関数の勾配ベクトルを計算する勾配ベクトル計算工程と、前記勾配ベクトルのノルムの値に基づいて第1の係数を計算する第1の係数計算工程と、該第1の係数に基づいて探索ベクトルを計算する探索ベクトル計算工程と、第2の係数を計算する第2の係数計算工程と、前記第2の係数に基づいて設計変数ベクトルを更新する設計変数ベクトル更新工程とを備え、前記第2の係数計算工程は前記第1の求解工程を含み、該第1の求解工程は勾配ベクトルに基づく反復解法であり、当該反復中は前記状態変数ベクトルを初期化しないことを特徴とする。

【0017】

【発明の実施の形態】

本実施形態の構成を図2に示す。図中、201はCPU、202は表示装置、203は入力装置、204は1次記憶装置、205は2次記憶装置、206は通信装置、207はバスラインである。

【0018】

本実施形態における各処理機能は、プログラムとして予め2次記憶装置205に格納されており、入力装置203或いは通信装置206等からのコマンド入力によって、1次記憶装置にロードされ、実行されるものである。

以下の説明のために対象とする問題の定式化を行う。

【0019】

有限要素法による定式化により、変関数は有限次元ベクトルで表現されるものとする、変関数の評価汎関数は、変数ベクトルの評価関数となる。以下、有限次元ベクトルで表現されているとして記述する。

状態変数ベクトル x 、設計変数ベクトル f をそれぞれ列ベクトルとして以下のよう

に書く：

$$(式1) \quad x = (x_1, x_2, \dots, x_m)^T$$

$$(式2) \quad f = (f_1, f_2, \dots, f_n)^T$$

ここで T は転置を表す。 x は m 次元ベクトル、 f は n 次元ベクトルである。

x および f の境界条件をそれぞれ B_1 、 B_2 とする：

$$(式3) \quad B_1(x) = 0$$

$$(式4) \quad B_2(f) = 0$$

状態変数ベクトル x および設計変数ベクトル f に関する評価関数を、それぞれ L_1 、 L_2 とする：

$$(式5) \quad L_1 = L_1(x; f)$$

$$(式6) \quad L_2 = L_2(f, x)$$

L_1 は x を変数ベクトル、 f をパラメタとする関数であり、 L_2 は x および f を変数ベクトルとする関数である。

【0020】

一般の最適化問題では、 K_1 個の等式拘束条件と K_2 個の不等式拘束条件とが課せられることが多い。そこで、設計変数に関する j 番目の等式拘束条件および k 番目の不等式拘束条件を、それぞれ Q_j 、 R_k とする：

$$(式7) \quad Q_j(f, x) = 0$$

$$(式8) \quad R_k(f, x) \geq 0$$

以上の表記より、最適設計問題は、拘束条件 (式4)、(式5)、(式6)、(式7)、および (式8) の下で、次式を満足する解として与えられる：

$$(式9) \quad \min [L_2]$$

ただし、状態変数 x は拘束条件 (式3) の下で次式を満足する解として得られる：

$$(式10) \quad \min [L_1]$$

上記定式化に基づき、本実施形態の処理を説明する。

以下では、説明を簡単にするために、(式7) の等式拘束条件を次式の1個とする：

【外1】

$$(式11) \quad Q(f) = \sum_{j=1}^{J=n} c_j f_t(j) + c = 0$$

【0021】

ここで c_j ($j = 0, 1, \dots, n$) は定数である。

【0022】

また (式8) の不等式拘束条件を次式のように、値の範囲に対するものとする：

$$(式12) \quad R_j^-(f) = f_t(j) - f_{j\min} \geq 0$$

$$R_j^+(f) = f_{j\max} - f_t(j) \geq 0$$

ただし、 $j = 1, 2, \dots, n$ 。

【0023】

従って、不等式拘束条件の数は設計変数の数の2倍となる。

【0024】

設計変数ベクトルと同次元のフラグベクトル a^+ 、および a^- を以下のように定義する:

$$\begin{aligned} a^+ &= 0, & \text{if } 0 \geq R_j^+ (f) \\ a^+ &= 1, & \text{otherwise} \end{aligned}$$

および

$$\begin{aligned} a^- &= 0, & \text{if } 0 \geq R_j^- (f) \\ a^- &= 1, & \text{otherwise} \end{aligned}$$

また、 a^+ および a^- が共に 1 である j の集合を Λ_1 、それ以外のインデクスの集合を Λ_2 とする。

【0025】

図 1 には、第二の求解工程のフローチャートが示されている。図中、ステップ S101 は、シミュレーション対象となる系の諸元を読み込む処理である。諸元の読み込みは、入力装置 203 或いは通信装置 206 からの入力データで行っても良いし、予め 2 次記憶装置 205 にファイルとして格納しておいたデータを読み出して利用することもできる。系の諸元には x 、 f の初期値、境界条件 B_1 、 B_2 、評価関数 L_1 、 L_2 、拘束条件 Q_j 、 R_k が含まれている。プログラムは、この情報によって、必要な変数領域を 1 次記憶装置 204 に確保し、値を設定する。

【0026】

ステップ S102 では、反復回数 t を 1 に初期化する。反復回数 t は、上述の第 2 の求解工程における第 1 の反復処理を行なった回数であり、以下のステップ S103 から 111 の処理の反復回数が設定値 T に達したら終了する。

【0027】

ステップ S103 では勾配ベクトル計算工程を実行する。同工程では、設計変数ベクトルに関する第 2 の評価関数のグラジエントベクトル g_t を計算する。 g_t は次式で計算される:

$$(式 13) \quad g_t = (\partial L_2 / \partial f_1, \partial L_2 / \partial f_2, \dots, \partial L_2 / \partial f_n)$$

T

$\partial L_2 / \partial f_j$ は解析的に解ける場合とそうでない場合がある。解析解がある

場合は、その計算工程を利用して（式10）を得る。そうでない場合は、自動微分という技術を利用して計算することが出来る。自動微分技術は、以下の〔文献6〕等によって公知である。

〔文献6〕久保田光一，伊理正夫：“アルゴリズムの自動微分と応用，”現代非線形科学シリーズ，コロナ社（1998年）

ステップS104では、（式13）で得られた g_t に対して、第一のベクトル修正工程を実行する。同工程は図6を用いて後述する。

【0028】

ステップS105では、収束判定工程を実行する。同工程では、 g_t のノルムの2乗を計算し、その値が0或いは0に近い非常に小さい値 ϵ 未満になると処理を終了する。ノルムの2乗は次式で計算する：

$$(\text{式14}) \quad \|g_t\|^2 = g_t^T g_t$$

ステップS106では、第一の係数計算工程を実行する。第一の係数計算工程では、（式15）により係数 β を計算する：

$$(\text{式15}) \quad \beta = \|g_t\|^2 / \|g_t - 1\|^2$$

ただし $t=1$ のときは $\beta=1$ とする。

【0029】

ステップS107では探索ベクトル計算工程を実行する。同工程では（式16）を用いて、反復回数 t 回目の探索ベクトル p_t を計算する：

$$(\text{式16}) \quad p_t \leftarrow g_{t-1} + \beta p_{t-1}$$

ステップS108では、（式16）より得られた p_t に対して、第一のベクトル修正工程を実行する。同工程は図6を用いて後述する。

【0030】

ステップS109ではライン探索工程を実行する。同工程では設計変数の更新のための係数 δ を決定する。同工程は図3を用いて後述する。

【0031】

ステップS110では設計変数ベクトル更新工程を実行する。同工程では設計変数ベクトルを更新する：

$$(\text{式17}) \quad f_{t+1} = f_t + \delta p_t$$

ステップ S111 では、設計変数ベクトルに対して第二のベクトル修正工程を実行する。同工程は図 7 を用いて後述する。

【0032】

ステップ S112 では t を $t+1$ に更新し、更新後の t が予め設定された反復回数を超えたら処理を終了する。

【0033】

以上が第二の求解工程の流れである。以下、各ステップについて詳述する。

【0034】

図 3 を参照してライン探索工程を説明する。同工程には、現在の設計変数値 f_t 、現在の設計変数値における探索ベクトル p_t が引数として与えられる。更に、評価関数や拘束条件は適宜与えられているとする。

【0035】

ステップ S301 では最大ステップサイズ Δ を、次式を用いて計算する：

【外 2】

$$(式18) \quad \Delta = \underset{j \in \Lambda_1}{\text{MIN}} [(ff(j) - f_t(j)) / p_t(j)]$$

【0036】

ただし

$$\begin{aligned} ff(j) &= f_{j \max}, \text{ if } p_t(j) > 0 \\ &= f_{j \min}, \text{ if } p_t(j) < 0 \end{aligned}$$

また、各変数の初期化を次式のように実行する：

$$\begin{aligned} (式19) \quad g_1 &= (3.0 - 5.0^{1/2}) / 2, \\ g_2 &= (5.0^{1/2} - 1) / 2, \\ d\Delta &= \Delta / T_m, \\ a_1 &= 0.0, \\ a_2 &= a_1 + d\Delta, \end{aligned}$$

ただし T_m は探索範囲を適当な区間に分割するための正整数であり、例えば 10 ～ 100 等の値を設定する。

【0037】

ステップ S302 では探索範囲絞り込み工程を実行する。同工程は図 4 を用いて後述する。

【0038】

ステップ S303 では探索範囲絞り込み工程で設定された δ と最大ステップサイズ Δ とを比較し、等しければ処理を終了する。そうでなければステップ S304 へ進む。

【0039】

ステップ S304 では極小点探索工程を実行する。同工程は図 5 を用いて後述する。以上でライン探索工程を終了する。

【0040】

図 4 を参照して探索範囲絞り込み工程を実行する。

【0041】

ステップ S401 では第 2 の評価関数値を計算する：

$$(\text{式 } 20) \quad L_{21} = L_2(x(f_t + a_1 p_t), f_t + a_1 p_t)$$

$$(\text{式 } 21) \quad L_{22} = L_2(x(f_t + a_2 p_t), f_t + a_2 p_t)$$

ただし $x(f_t + a_1 p_t)$ 、および $x(f_t + a_2 p_t)$ は図 10 を用いて後述する第一の求解工程を実行することによって得られる。

【0042】

ステップ S402 では L_{21} と L_{22} とを比較し、 L_{21} が大きければ処理を終了し、そうでなければステップ S403 へ進む。

【0043】

ステップ S403 では t を 0 に、 $\mu(t)$ を 0 に初期化する。

【0044】

ステップ S404 では $\mu(t)$ を更新する：

$$(\text{式 } 22) \quad \mu(t) = \mu(t-1) + d\Delta$$

ステップ S405 では L_{21} と L_{22} を更新する：

$$(\text{式 } 23) \quad L_{21} = L_{22}$$

$$(\text{式 } 24) \quad L_{22} = L_2(x(f_x + \mu(t) p_t), f_x + \mu(t) p_t)$$

ただし $x(f_x + \mu(t) p_t)$ は図 12 を用いて後述する第一の求解工程を実

行することによって得られる。

【0045】

ステップS406ではL21とL22とを比較し、L21がL22以下ならステップS408へ、そうでなければステップS407へ進む。

【0046】

ステップS407では、 t を $t+1$ に更新したあと、 t と T_m を比較し、 t が大きければステップS409へ、そうでなければステップS404へ進む。

【0047】

ステップS408では、次式を実行する：

$$(\text{式 } 25) \quad a_1 = \mu(t) - 2d\Delta$$

$$a_2 = \mu(t)$$

ステップS409では $\mu(t)$ と Δ を比較し、 $\mu(t)$ が大きければ δ に Δ を代入し、そうでなければ、 δ に Δ 以外の任意の値を代入して、探索範囲絞込み工程を終了する。

【0048】

図5を用いて極小点探索工程を説明する。

【0049】

ステップS501では次式を計算し

$$(\text{式 } 26) \quad da = a_2 - a_1$$

da が予め設定された微小な正の実数以下であればステップS504へ進み、そうでなければステップS502へ進む。

【0050】

ステップS502では次式を計算する：

$$(\text{式 } 27) \quad v_1 = a_1 + g_1 da$$

$$v_2 = a_1 + g_2 da$$

更に探索範囲の絞込みを行う：

$$(\text{式 } 28) \quad a_2 = v_2, \quad \text{if } L_{21} < L_{22}$$

$$a_1 = v_1, \quad \text{otherwise}$$

これは、黄金分割法による区間縮小である。

【0051】

ステップS503では、以下の式により第二の評価関数の値を計算し、ステップS501へ進む。

【0052】

$$(\text{式 } 29) \quad L21 = L2(x(f_x + a1p_t), f_x + a1p_t)$$

$$L22 = L2(x(f_x + a2p_t), f_x + a2p_t)$$

ただし $x(f_t + a1p_t)$ 、および $x(f_t + a2p_t)$ は図10を用いて後述する第一の求解工程を実行することによって得られる。

【0053】

ステップS504では探索区間の中点 a_c と、その位置での第2の評価関数値 $L2c$ を計算する：

$$(\text{式 } 30) \quad a_c = (a1 + a2) / 2$$

$$(\text{式 } 31) \quad L2c = L2(x(f_x + a_cp_t), f_x + a_cp_t)$$

ステップS505では δ の値を決定する：

$$\begin{aligned} (\text{式 } 32) \quad \delta &= a_c, \text{ if } L2c = \text{MIN}[L21, L22, L2c, L2e] \\ &= a1, \text{ else if } L21 = \text{MIN}[L21, L22, L2c, L2e] \\ &= a2, \text{ else if } L22 = \text{MIN}[L21, L22, L2c, L2e] \\ &= \Delta, \text{ else if } L2e = \text{MIN}[L21, L22, L2c, L2e] \\ &= 0, \text{ otherwise} \end{aligned}$$

ただし $\text{MIN}[]$ は引数の中の最小値を返す関数である。以上で極小点探索工程を終了する。

【0054】

ステップS104およびステップS108で実施される第一のベクトル修正工程について、図6を用いて説明する。

【0055】

ステップ S601 では第二の係数計算工程を実行する。ここでは、単位法線ベクトル NN 、原点から超平面への垂線の長さ DST 、および係数 B の値を計算する。同処理は図 8 を用いて詳述する。

【0056】

ステップ S602 では第一のベクトル射影工程を実行する。この工程は、ステップ S601 で計算した単位法線ベクトル NN 、原点から超平面への垂線の長さ DST 、及び係数 B を使用し、次式に基づいてベクトル X を修正する：

$$\begin{aligned} \text{(式 33)} \quad X(j) &= X(j) - B \quad NN(j), \quad \text{if } j \in \Lambda_1 \\ &= 0, \quad \text{otherwise} \end{aligned}$$

以上で第一のベクトル修正工程を終了する。

【0057】

ステップ S111 で実施される第二のベクトル修正工程について、図 7 を用いて説明する。

【0058】

ステップ S701 では第二の係数計算工程を実行する。同処理は図 8 を用いて詳述する。

【0059】

ステップ S702 では第二のベクトル射影工程を実行する。この工程は、ステップ S701 で計算した単位法線ベクトル NN 、原点から超平面への垂線の長さ DST 、及び係数 B を使用し、次式に基づいてベクトル X を修正する：

$$\begin{aligned} \text{(式 34)} \quad X(j) &= X(j) - B \quad NN(j), \quad \text{if } j \in \Lambda_1 \\ &= f_{j \max}, \quad \text{if } a^+ = 0 \\ &= f_{j \min}, \quad \text{if } a^- = 0 \end{aligned}$$

以上で第二のベクトル修正工程を終了する。

【0060】

図 8 を用いて第二の係数計算工程を説明する。

【0061】

ステップ S801 では法線ベクトル計算工程を実行し、 NN 、および DST を得る。同工程は図 9 を用いて後述する。

【0062】

ステップS802では、次式を用いて係数Bを計算する：

【外3】

$$(式35) \quad B = \sum_{k \in \Lambda_1} u(j) N N(j) - DST$$

【0063】

以上で第二の係数計算工程を終了する。

【0064】

図9を用いてステップS801の法線ベクトル計算工程を説明する。本工程では、単位法線ベクトルNNと、原点から超平面への垂線の長さDSTを計算する。

【0065】

ステップS901では次式の変数を計算する：

$$(式36) \quad DST = -c(0)'$$

ただし $c(0)'$ は次式で計算される：

【外4】

$$(式37) \quad c(0)' = c(0) + \sum_{k \in \Lambda_2} c(k) f_t(k)$$

$$(式38) \quad D = \left(\sum_{k \in \Lambda_1} c(k)^2 \right)^{-1/2}$$

$$(式39) \quad \sum_{j \in \Lambda_1} c(j) f_t(j) + c(0)' = 0$$

【0066】

(式36)のDSTを次式で修正する：

$$(式40) \quad DST = DST / D$$

上式で与えられるDSTが、原点から超平面への垂線の長さとなる。

【0067】

ステップS902では法線ベクトルの計算を実行する。

【0068】

まず、正規化されていない法線ベクトル N を (式 4 1) および (式 4 2) より計算する:

【外 5】

$$(式41) \quad N(j) = \prod_{k \in \Lambda_1} c(k)/c(j), \text{ if } j \in \Lambda_1$$

【0069】

$$(式42) \quad N(j) = f_t(j), \quad \text{otherwise}$$

(式 4 1) を用いて、部分空間に対する単位法線ベクトル NN を次式で計算する:

【外 6】

$$(式43) \quad NN(j) = N(j) \left(\sum_{j \in \Lambda_1} \prod_{k \in \Lambda_1} c(k)^2 / c(i)^2 \right)^{-1/2}, \text{ if } j \in \Lambda_1$$

【0070】

例えば、等式拘束条件として設計変数ベクトルの要素の総和が 1 である場合は、 $c(0) = -1$, $c(j) = 1$, $j = 1, 2, \dots, n$ であるので、単位法線ベクトルの成分は次のように簡単になる:

$$(式44) \quad NN(j) = |\Lambda_1|^{-1/2}, \text{ if } j \in \Lambda_1$$

ただし $|\Lambda_1|$ は Λ_1 の要素数である。

【0071】

以上で法線ベクトル計算工程を終了する。

【0072】

図 10 を用いて、第一の求解工程を説明する。

【0073】

構造が変化すると特性関数値が 0 となる要素が出現する可能性がある。このような構造に対して構想解析を行うと、係数行列が正則でなくなるので、直接法では解けなくなる。そのために、第 1 の求解工程として、以下の 3 通りの方法を用いる:

(1) 値が 0 の特性関数値は 0 に近い微小な値に置き換えて、直接法で連立 1 次方程式を解く。

【0074】

(2) 値が0を超える要素のみで連立1次方程式を構成し直して、直接法で解く。

【0075】

(3) 逆行列の計算を必要としない、勾配ベクトルを用いた反復法で解く。これには最急降下法、共役勾配法等が含まれる。

【0076】

以下では共役勾配法を用いた求解工程を説明する。

【0077】

ステップS1001では、シミュレーション対象となる系の諸元を読み込む。系の諸元にはxの初期値、設計パラメタfの値、境界条件B₁、評価関数L₁が含まれている。プログラムは、この情報によって、必要な変数領域を1次記憶装置204に確保し、値を設定する。

【0078】

ステップS1002では反復回数tを1に初期化する。反復回数tは、上述の第一の求解工程に対する第1の反復処理に対するものであり、以下のステップS1003からステップS1010の処理の反復回数が設定値を超えたら終了する。

【0079】

ステップS1003では状態変数ベクトルに関する第一の評価関数のグラジエントベクトルg_{1, t}を計算する。g_{1, t}は次式で計算される：

$$\text{(式45)} \quad g_{1, t} = (\partial L_1 / \partial x_1, \partial L_1 / \partial x_2, \dots, \partial L_1 / \partial x_n)^T$$

$\partial L_1 / \partial x_j$ は解析的に解ける場合とそうでない場合がある。解析解がある場合は、その計算工程を利用して(式45)を得る。そうでない場合は、自動微分という技術を利用して計算することが出来る。自動微分技術は〔文献6〕等によって公知である。

【0080】

ステップS1004ではg_{1, t}のノルムの2乗を計算し、その値が0或いは

0に近い非常に小さい値のときに処理を終了する。ノルムの2乗は次式で計算する:

$$(式46) \quad \|g_{1,t}\|^2 = g_{1,t}^T g_{1,t}$$

ステップS1005では、(式123)より β を計算する:

$$(式47) \quad \beta = \|g_{1,t}\|^2 / \|g_{1,t-1}\|^2$$

ただし $t=1$ のときは $\beta=1$ とする。

【0081】

ステップS1006では(式124)を用いて、探索方向ベクトル p_t を計算する:

$$(式48) \quad p_{1,t} = -g_{1,t-1} + \beta p_{1,t-1}$$

ステップS1007ではライン探索工程を実行する。設計変数の更新のための係数 α を決定するものである。構造解析問題を有限要素モデルを用いて解く場合には、 α の値は次式で与えられる:

$$(式49) \quad \alpha = p_{1,t}^T g_{1,t} / (p_{1,t}^T A p_{1,t})$$

ステップS1008では状態変数ベクトルを更新する:

$$(式50) \quad x_{t+1} = x_t + \alpha p_{1,t}$$

ステップS1009では t を $t+1$ に更新し、更新後の t が予め設定された反復回数を超えたら処理を終了する。

【0082】

以上で第一の求解工程を終了する。

【0083】

(実施例)

本実施例では、任意の位置に加重を受ける片持ち梁の最適形状自動設計に本発明を適用するものである。説明を簡単にするために平面歪問題に限定する。

【0084】

図11に示すように、構造部材の存在を可能とする設計領域は、長方形1102であり、有限要素法に従って、該領域を縦 n_y 、横 n_x に等間隔に分割する。分割にされた部分領域をセルと呼び、左下および右上のセルが $(1, 1)$ (n_y, n_x)となるように番号付けを行う。同様に格子点をノードと呼び、左下およ

び右上のノードが $(1, 1) \dots (n_y + 1, n_x + 1)$ となるように番号付けを行う。

【0085】

図中 1101 は支持部材、1103 は荷重ベクトルである。セル (j, k) には特性関数値 $f(j, k)$ が対応する。ここで特性関数値とはセル (j, k) における構造部材の存在確率を示す 0 から 1 の正の実数値をとる変数であり、本発明における設計変数ベクトル f の要素である：

$$(式 5 1) \quad f = (f(1, 1), f(1, 2), \dots, f(n_y, n_x))^T$$

同様にノード (j, k) には横方向変位 $u(j, k)$ と縦方向変位 $v(j, k)$ が対応する。これらは任意の値を取る実数であり、本発明における状態変数ベクトル U の要素である：

$$(式 5 2) \quad U = (u(1, 1), v(1, 1), u(1, 2), v(1, 2), \dots, u(n_y + 1, n_x + 1), v(n_y + 1, n_x + 1))^T$$

同様に剛性マトリクスを A 、加重ベクトルを b と書けば、有限要素法の良く知られた結果として、状態変数ベクトル U は、次式で与えられる評価汎関数の最適化問題の解として与えられる：

$$(式 5 3) \quad L_1 = (1/2) U^T A U - b^T U$$

より具体的には、以下の線型方程式の解として与えられることが知られている：

$$(式 5 4) \quad A U - b = 0$$

第一の求解工程において計算される g_1, t は (式 5 4) の左辺として与えられる。(式 5 4) は連立一次方程式であるので、求解法として、通常、逆行列に基づく直接解法が採用される。しかし構造最適化問題においては、剛性マトリクス A は設計変数ベクトル f の関数であり、 f に 0 の値を取る要素があれば、剛性マトリクス A はランク落ちし正則でなくなってしまう。その結果直接解法で解けないという事態に陥るので、本発明で共役勾配法を採用しているのは、このような理由による。

【0086】

設計変数ベクトル f に対する評価関数 L_2 は総歪エネルギーで定義される。

【0087】

$$(式 55) \quad L_2 = (1/2) U^T A U$$

第二の求解工程で必要となる g_t は次式で計算できる:

$$(式 56) \quad g_t = \partial L_2 / \partial f(e) = - (1/2) U_e^T A_e U_e$$

ただし U_e 、 A_e は、それぞれ、要素 e に対応するノードにおける変位ベクトル、該変位ベクトルに対応する要素剛性マトリクスである。

【0088】

図 12 に本実施例に対する計算結果を示す。図中、黒い領域が構造部材が存在する領域である。

【0089】

本実施例において、構造部材の設計領域のアスペクト比は縦／横が 2 対 1 であり、その解析解は、水平方向に対して ± 45 度の梁が組み合わされたものであることが知られている。

【0090】

図 12 に示した計算結果は、このような解析解とよく一致していることがわかる。

【0091】

尚、本発明は、単一の機器からなる装置に適用しても、複数の機器から構成されるシステムに適用してもよい。また、上述した実施形態の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、装置あるいはシステムに供給し、装置あるいはシステム内のコンピュータが記憶媒体に格納されたプログラムコードを読み出して実行することによって達成してもよい。

【0092】

更に、装置あるいはシステム内のコンピュータが記憶媒体に格納されたプログラムコードを読み出して実行することによって、上述した実施形態の機能を直接実現するばかりでなく、そのプログラムコードの指示に基づいて、コンピュータ上で稼動している OS などの処理により、上述の機能を実現される場合も含まれる。

【0093】

これらの場合、そのプログラムコードを記憶した記憶媒体は本発明を構成する

ことになる。

【0094】

以下、上記実施形態に係る本発明の特徴を整理する。

【0095】

特徴 1.

設計変数ベクトルをパラメタとして状態変数ベクトルに対する第 1 の評価関数の最適化問題を解く第 1 の求解工程と、

前記第 1 の求解工程で求められた状態変数ベクトルと、前記設計変数ベクトルとに対する第 2 の評価関数の最適化問題を解く第 2 の求解工程とを備える最適設計方法において、

前記第 2 の求解工程は、

前記設計ベクトルに関する前記第 2 の評価関数の勾配ベクトルを計算する勾配ベクトル計算工程と、

前記勾配ベクトルのノルムの値に基づいて第 1 の係数を計算する第 1 の係数計算工程と、

該第 1 の係数に基づいて探索ベクトルを計算する探索ベクトル計算工程と、

第 2 の係数を計算する第 2 の係数計算工程と、

前記第 2 の係数に基づいて設計変数ベクトルを更新する設計変数ベクトル更新工程とを備え、

前記第 2 の係数計算工程は前記第 1 の求解工程を含み、該第 1 の求解工程は勾配ベクトルに基づく反復解法であり、当該反復中は前記状態変数ベクトルを初期化しないことを特徴とする最適設計方法。

【0096】

特徴 2.

前記ライン探索工程は、

前記拘束条件に基づいて最大ステップサイズを決定する最大ステップサイズ計算工程と、

前記最大ステップサイズより決定される探索区間の中から極小点を決定する探索範囲絞込み工程と、

極小点における前記第 2 の評価関数値と探索範囲の終点における前記第 2 の評価関数値との比較に基づいて極小点を決定する極小点決定工程とを備えることを特徴とする特徴 1 に記載の最適設計方法。

【0 0 9 7】

特徴 3.

前記第 2 の求解工程は、前記設計変数ベクトルより、前記第 2 の評価関数に寄与しない構造要素に対応する成分を消去する消去工程を含むことを特徴とする特徴 1 に記載の最適設計方法。

【0 0 9 8】

特徴 4.

前記消去工程では、前記設計変数ベクトルの各成分の値を増減させたとき、当該設計変数ベクトルに関する前記第 2 の評価関数の感度の値が常に 0 である成分を消去することを特徴とする特徴 1 に記載の最適設計方法。

【0 0 9 9】

特徴 5.

前記消去工程は、前記第 2 の求解工程で計算される感度ベクトルが 0 である成分に対応する要素であって、かつ構造要素の存在率が 0 でない要素に対して行われることを特徴とする特徴 1 に記載の最適設計方法。

【0 1 0 0】

特徴 6.

前記消去工程は、前記第 2 の求解工程の反復処理の所定回に 1 回行われることを特徴とする特徴 1 に記載の最適設計方法。

【0 1 0 1】

特徴 7.

前記設計変数ベクトルは、各要素における構造部材の存在率であることを特徴とする特徴 1 に記載の最適設計方法。

【0 1 0 2】

特徴 8.

前記勾配ベクトル計算工程により計算された勾配ベクトルを拘束条件に基づい

て修正する勾配ベクトル修正工程を備えることを特徴とする特徴 1 に記載の最適設計方法。

【0103】

特徴 9.

前記探索ベクトル計算工程により計算された探索ベクトルを拘束条件に基づいて修正する探索ベクトル修正工程を備えることを特徴とする特徴 1 に記載の最適設計方法。

【0104】

特徴 10.

前記設計変数ベクトル更新工程により更新された設計変数ベクトルを拘束条件に基づいて修正する設計変数ベクトル修正工程を備えることを特徴とする特徴 1 に記載の最適設計方法。

【0105】

特徴 11.

前記勾配ベクトルのノルムの値に基づいて前記第 2 の求解工程を終了するか否かを判定する収束判定工程を備えることを特徴とする特徴 1 に記載の最適設計方法。

【0106】

特徴 12.

設計変数ベクトルをパラメタとして状態変数ベクトルに対する第 1 の評価関数の最適化問題を解く第 1 の求解手段と、

前記第 1 の求解手段で求められた状態変数ベクトルと、前記設計変数ベクトルとに対する第 2 の評価関数の最適化問題を解く第 2 の求解手段とを備える最適設計装置において、

前記第 2 の求解手段は、

前記設計ベクトルに関する前記第 2 の評価関数の勾配ベクトルを計算する勾配ベクトル計算手段と、

前記勾配ベクトルのノルムの値に基づいて第 1 の係数を計算する第 1 の係数計算手段と、

該第 1 の係数に基づいて探索ベクトルを計算する探索ベクトル計算手段と、
第 2 の係数を計算する第 2 の係数計算手段と、
前記第 2 の係数に基づいて設計変数ベクトルを更新する設計変数ベクトル更新手段とを備え、

前記第 2 の係数計算手段は前記第 1 の求解手段を含み、該第 1 の求解手段は勾配ベクトルに基づく反復解法を行ない、当該反復中は前記状態変数ベクトルを初期化しないことを特徴とする最適設計装置。

【0107】

特徴 13.

設計変数ベクトルをパラメタとして状態変数ベクトルに対する第 1 の評価関数の最適化問題を解く第 1 の求解工程と、

前記第 1 の求解工程で求められた状態変数ベクトルと、前記設計変数ベクトルとに対する第 2 の評価関数の最適化問題を解く第 2 の求解工程とを備える最適設計プログラムにおいて、

前記第 2 の求解工程は、

前記設計ベクトルに関する前記第 2 の評価関数の勾配ベクトルを計算する勾配ベクトル計算工程と、

前記勾配ベクトルのノルムの値に基づいて第 1 の係数を計算する第 1 の係数計算工程と、

該第 1 の係数に基づいて探索ベクトルを計算する探索ベクトル計算工程と、

第 2 の係数を計算する第 2 の係数計算工程と、

前記第 2 の係数に基づいて設計変数ベクトルを更新する設計変数ベクトル更新工程とを備え、

前記第 2 の係数計算工程は前記第 1 の求解工程を含み、該第 1 の求解工程は勾配ベクトルに基づく反復解法であり、当該反復中は前記状態変数ベクトルを初期化しないことを特徴とする最適設計プログラム。

【0108】

【発明の効果】

以上説明したように、本発明によれば、個別の問題に依存する経験則を利用す

ること無しに、トポロジーを含む構造最適設計を実行することができる。

【図面の簡単な説明】

【図 1】

第 2 の求解工程の処理手順を示すフローチャートである。

【図 2】

本実施形態に係る装置のブロック構成図である。

【図 3】

ライン探索工程の処理手順を示すフローチャートである。

【図 4】

探索範囲絞込み工程の処理手順を示すフローチャートである。

【図 5】

極小点探索工程の処理手順を示すフローチャートである。

【図 6】

第 1 のベクトル修正工程の処理手順を示すフローチャートである。

【図 7】

第 2 のベクトル修正工程の処理手順を示すフローチャートである。

【図 8】

係数 B の計算工程の処理手順を示すフローチャートである。

【図 9】

法線ベクトル計算工程の処理手順を示すフローチャートである。

【図 10】

第 1 の求解工程の処理手順を示すフローチャートである。

【図 11】

実施例の問題設定の説明図である。

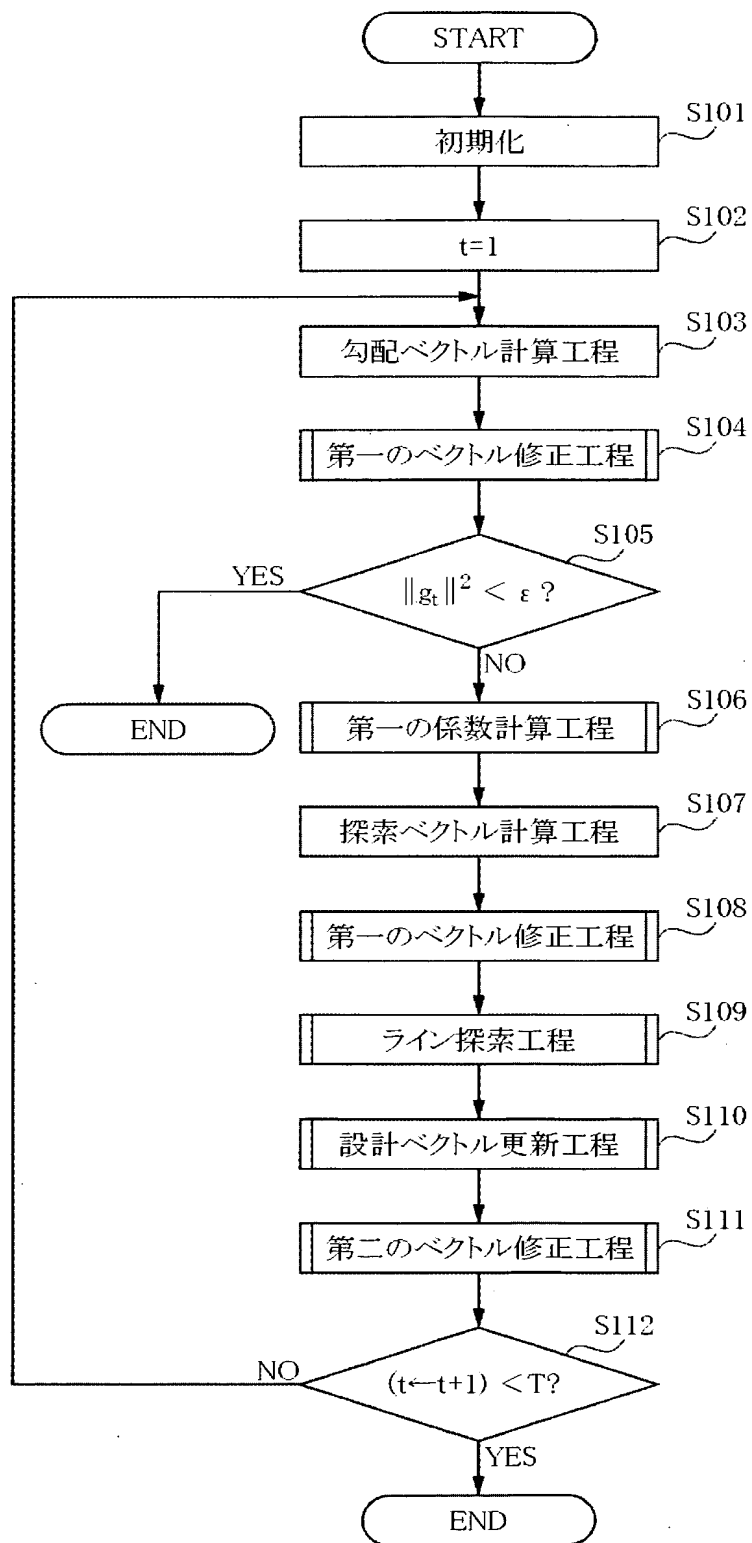
【図 12】

実施例の計算結果を示す図である。

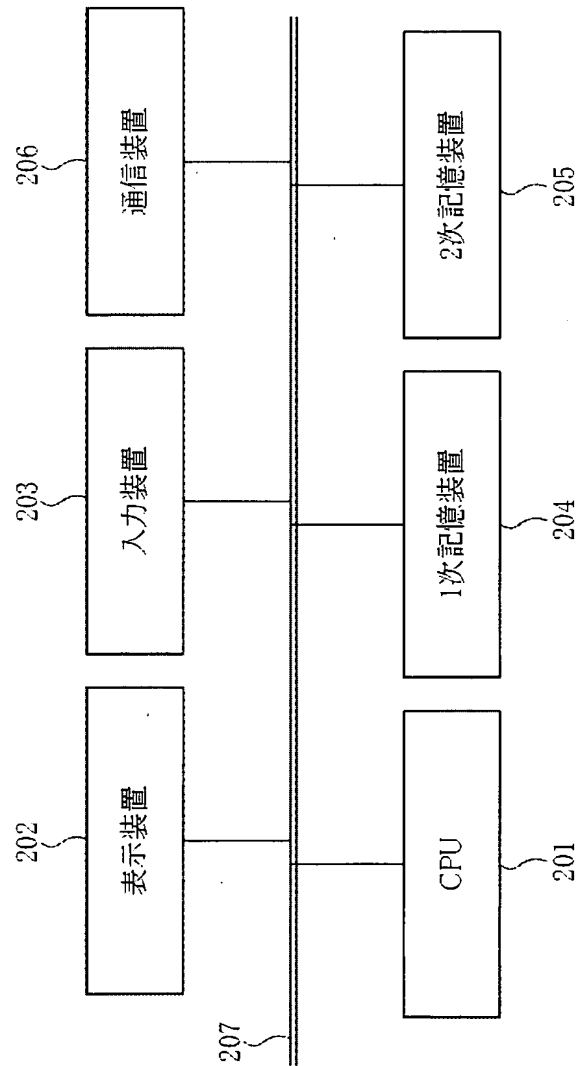
【書類名】

図面

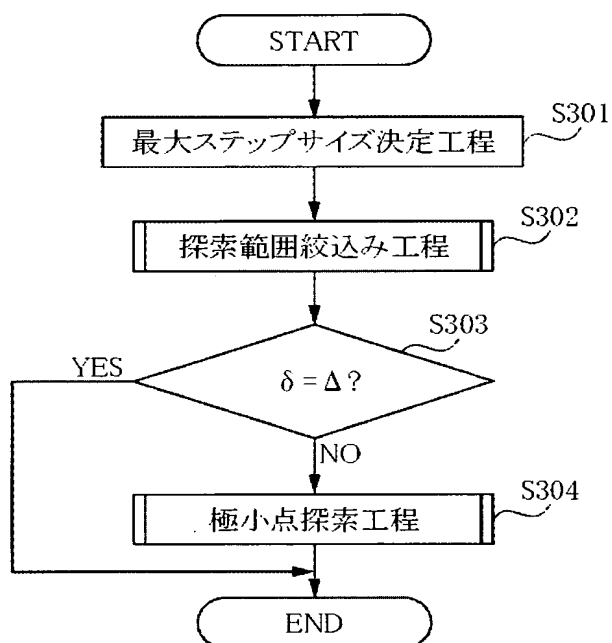
【図 1】



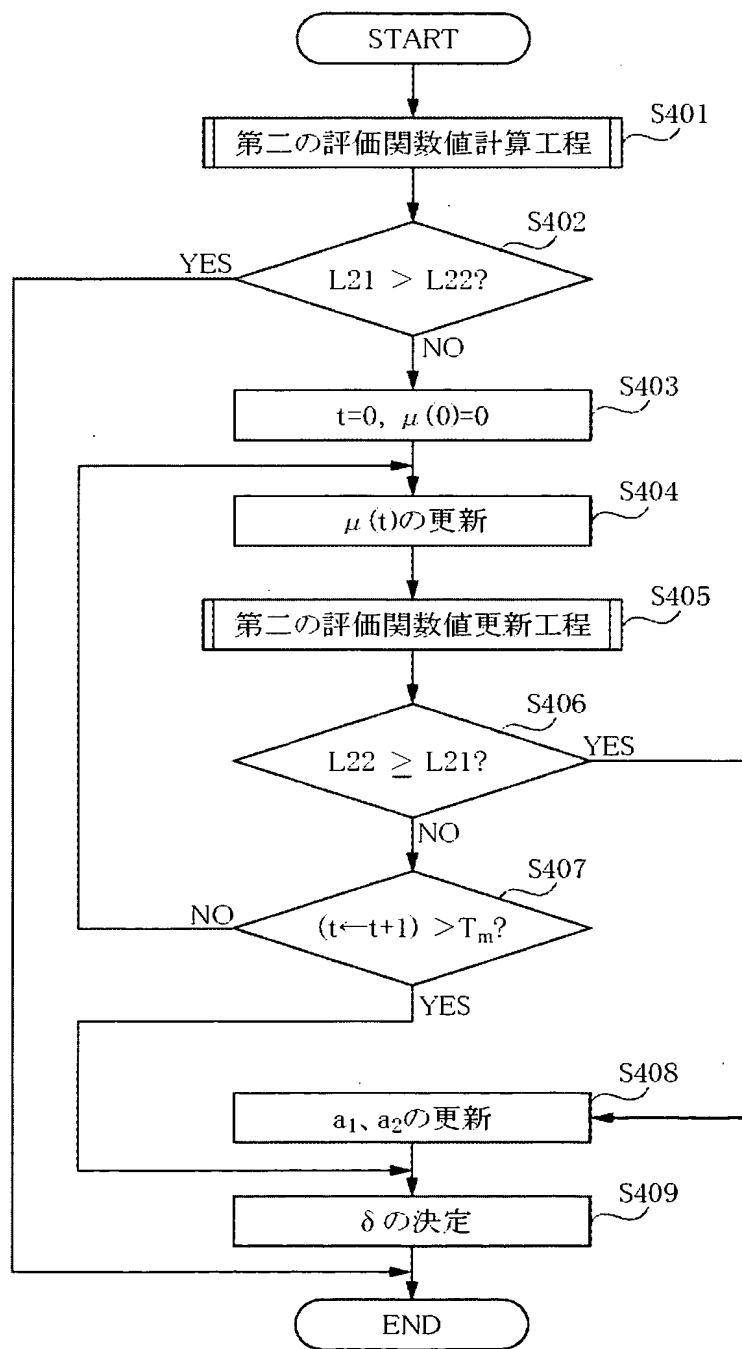
【図 2】



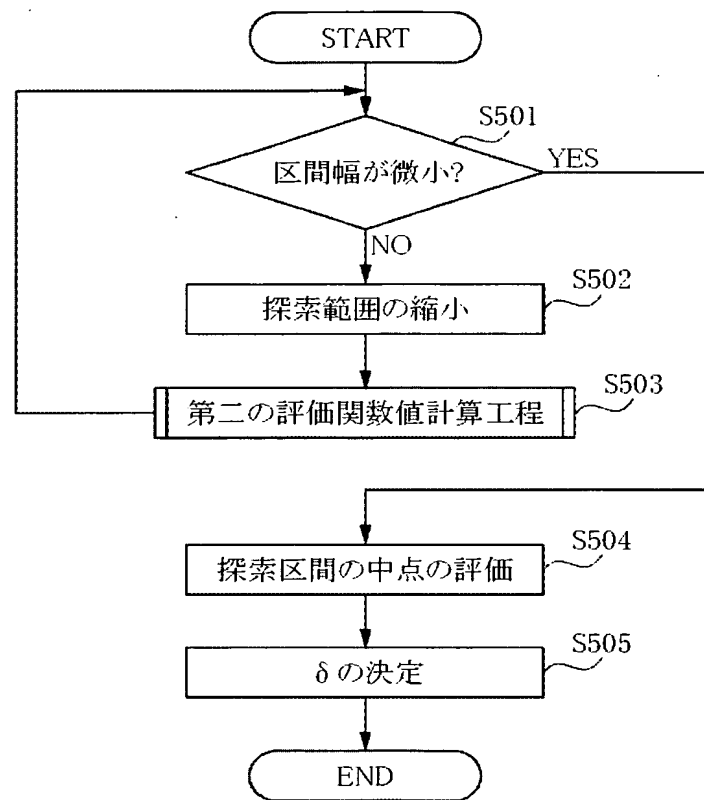
【図 3】



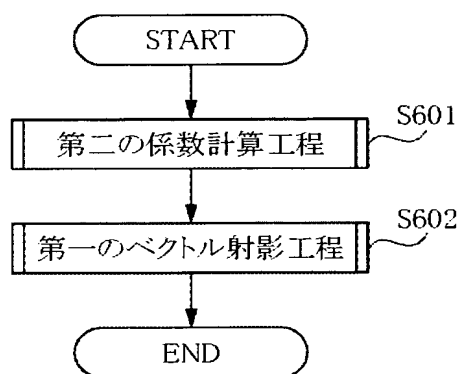
【図 4】



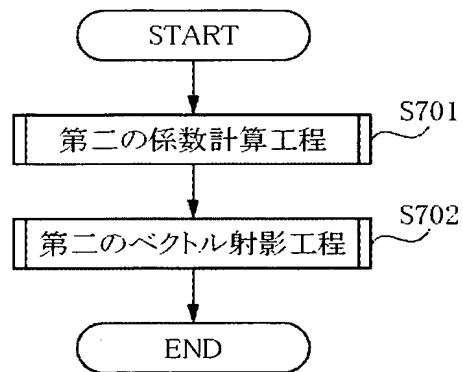
【図 5】



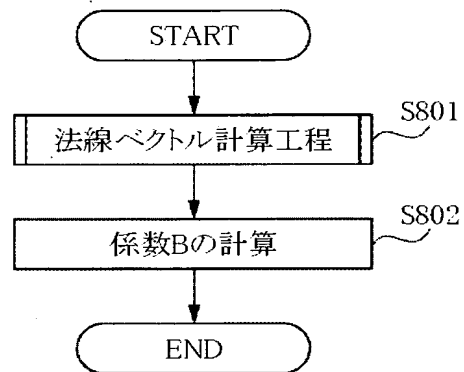
【図 6】



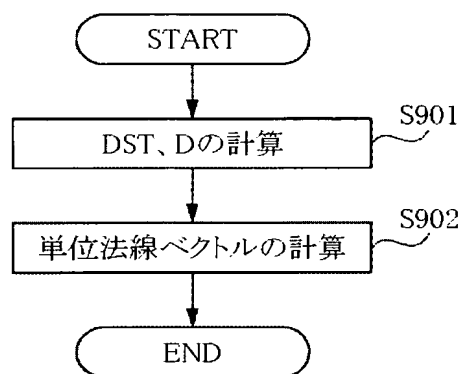
【図 7】



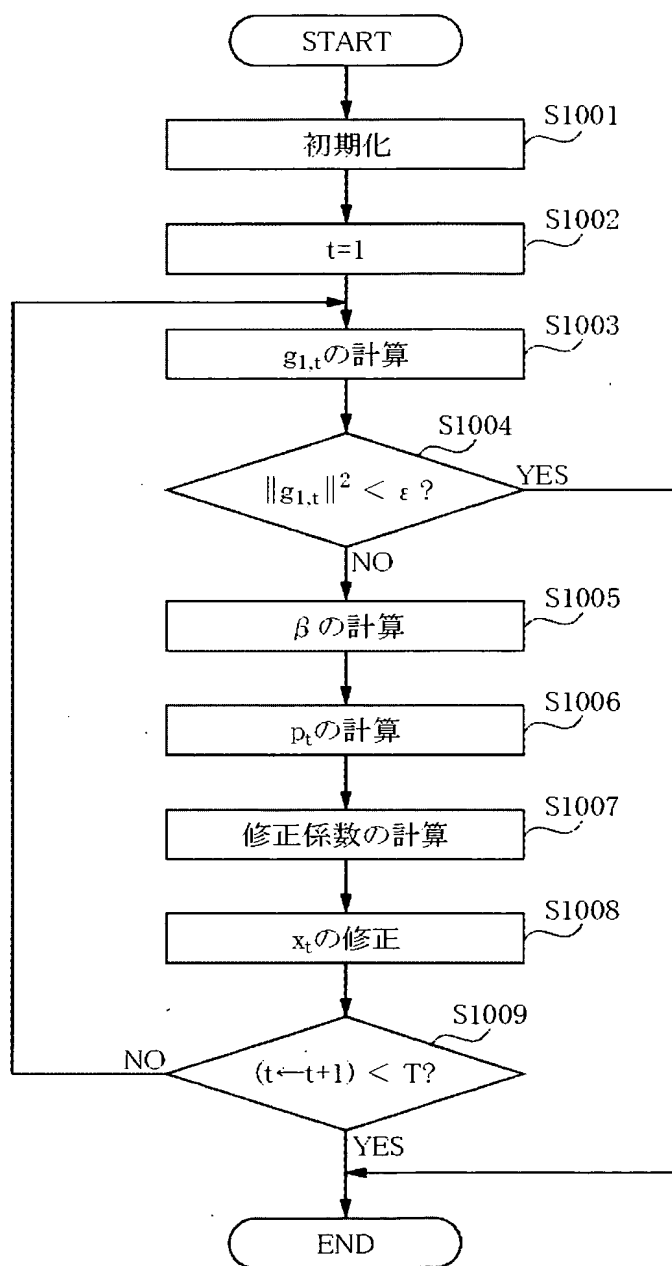
【図 8】



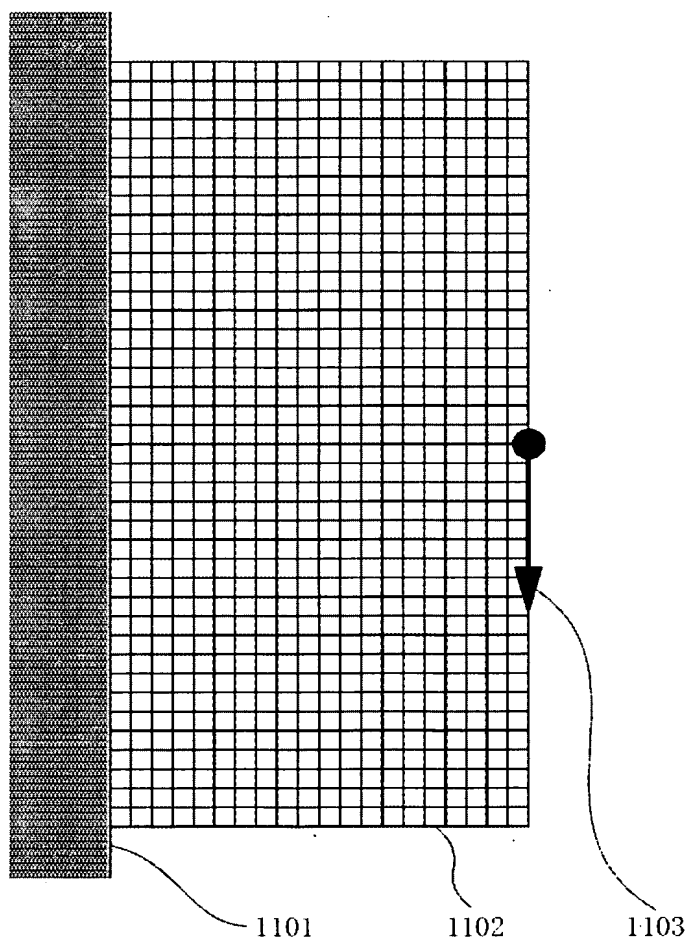
【図 9】



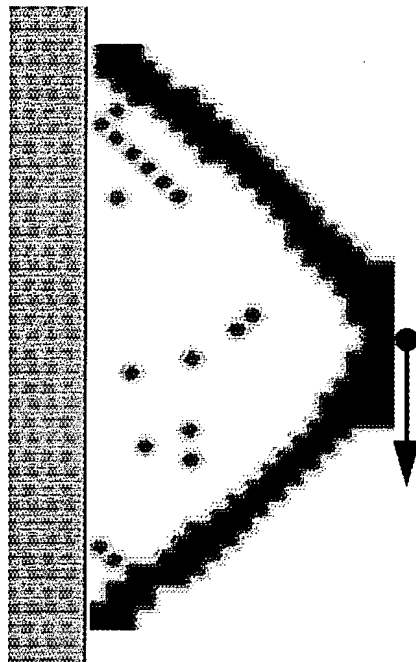
【図 10】



【図 11】



【図 12】



【書類名】 要約書

【要約】

【課題】 最適設計方法において構造変化に正確に対応する。

【解決手段】 設計変数ベクトルをパラメタとして状態変数ベクトルに対する第 1 の評価関数の最適化問題を解く第 1 の求解工程と、求められた状態変数ベクトルと、設計変数ベクトルとに対する第 2 の評価関数の最適化問題を解く第 2 の求解工程とを備え、第 2 の求解工程は、設計ベクトルに関する第 2 の評価関数の勾配ベクトルを計算する工程（S 1 0 3）と、勾配ベクトルのノルムの値に基づいて第 1 の係数を計算する工程（S 1 0 6）と、第 1 の係数に基づいて探索ベクトルを計算する工程（S 1 0 7）と、第 2 の係数を計算する工程（S 1 0 9）と、第 2 の係数に基づいて設計変数ベクトルを更新する工程（S 1 1 0）とを備え、第 2 の係数計算工程は第 1 の求解工程を含み、該第 1 の求解工程は勾配ベクトルに基づく反復解法であり、反復中は状態変数ベクトルを初期化しない。

【選択図】 図 1

特願 2 0 0 2 - 3 6 3 9 1 7

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 1 0 0 7]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社